REMARKS

Applicant is in receipt of the Office Action mailed January 23, 2004. Claims 1, 4-6, 8, 10, 12-14, 17-19, 21, 23, 25-27, 30-32, 34, 36, 38-40, 43, and 44 have been amended to more clearly claim the invention. New claims 46 and 47 have been added. Thus, claims 1, 3-14, 16-27, 29-40, and 42-47 remain pending in the case. Further consideration of the present case is earnestly requested in light of the following remarks.

Section 102 Rejections

Claims 1, 3-5, 7-14, 16-18, 20-27, 29-31, 33-40, and 42-45 were rejected under 35 U.S.C. 102(e) as being anticipated by Blowers et al. (U.S. Patent 6,298,474 B1, "Blowers"). Applicant respectfully traverses the rejection.

Amended claim 1 recites:

1. (Currently Amended) A method for generating a computer program, the method comprising:

receiving user input specifying a prototype, wherein the prototype comprises a series of functional operations, wherein at least one of the operations has an associated one or more parameters;

automatically generating a program that implements the prototype, in response to the specified prototype;

wherein said automatically generating the program comprises automatically generating a graphical user interface for the program;

wherein said automatically generating the graphical user interface comprises automatically creating graphical user interface elements associated with the one or more parameters, wherein during execution of the program, the graphical user interface elements are displayed and are operable to receive user input and/or display output.

Applicant has carefully reviewed the Blowers patent and believes that Blowers does not teach or suggest at least the above recited limitations of amended claim 1. As noted in the previous Response, Blowers refers to the user's creation of the control flow

structure, which presumably specifies the machine vision algorithm the user desires to create, but does not relate to a graphical user interface, and specifically does not disclose automatically creating a graphical user interface for the program.

Applicant notes that Figures 7 and 8 of Blowers illustrate a screen display that is used by a user to create a task or sequence, i.e., Figures 7 and 8 illustrate a task sequencer list. The screen displays in Figures 7 and 8 of Blower are used by the user to specify steps in a sequence and configure parameters in the various tasks the user is creating. For example, as Blowers describes in col. 8, lines 64-67, "Graphical representations or icons are selected from the tool boxes of FIG. 5 which correspond to desired functional tasks and are linked into the tree structure of FIG. 6 by a task sequencer interface 50 in the desired locations."

Applicant respectfully submits that the Examiner has incorrectly equated Applicant's *graphical user interface elements* with Blowers' controls. Applicant notes that nowhere does Blowers teach or suggest, or even mention, a graphical user interface (GUI) for the program, and specifically does not teach or suggest automatically generating a program, including automatically generating a graphical user interface for the automatically generated program. While Blowers does illustrate a GUI in Figures 7-9, Applicant notes that the GUIs of Figures 7 and 8 are "parameter pages" for configuring tasks in the sequence of tasks, and are thus not graphical user interfaces for the program itself. For example, as Blowers states in col. 9, lines 7-10, "Parameters are configurable on each of the tasks where the parameters control the way the function generates results, as illustrated in FIGS. 7 and 8 with respect to the blob and alignment vision tools, respectively".

The Blowers patent uses the term "control" to refer to a type of software program, e.g., an Active X control. In the Blowers context, the term "control" refers to a particular type of software program corresponding to a certain control or object standard such as Active X. Thus, for example, the cited portion of Blowers at column 3 lines 1-5 refers to standard controls such as Active X controls. These Active X controls are discussed as being able to provide inputs or outputs to other tools or other Active X controls. Thus, the controls referred to in Blowers, e.g., at column 3 lines 24-30, do not refer to graphical user interface elements, but rather refer to a certain type of software program, as is well

known to those skilled in the art of programming. Thus the reference to "controls" in Blowers does not teach *the user interface elements* recited in the present claims.

Thus, as noted previously, column 3 lines 1-5 and 24-30 of Blowers uses the term "controls" in a different manner than the "user interface controls" described in the present application. In the present application, the term "user interface controls" refers to a *graphical user interface element* which may be included in a graphical user interface which enables the user to provide input to a program and/or display output from a program during program execution. The claims have been amended to emphasize this distinction. Support for these amendments may be found in the specification at page 8 lines 25-31, which notes that "the user interface control for a given input parameter is operable to receive user input, such as during program execution, and provide this input to the associated operation in the program. Similarly, the user interface control for a given output parameter is operable to receive output data from the program such as *during program execution*, and display this output to the user." Furthermore, on page 9, lines 5-8, the specification states "the term "user interface control" may refer to an aggregate of simpler user interface controls or *user interface elements* which together form a suitable means for receiving an input parameter value or displaying an output parameter value". Thus, the present application utilizes the term "user interface control" to refer to a *graphical user interface element* in a graphical user interface for receiving user input and/or displaying output to the user during program execution.

Applicant further notes that the GUI of Figure 9 is for the runtime engine in general, and that results of the program execution are displayed in "a rolling results window of FIG. 9 through a results interface 60" (col. 9, lines 23-25). Applicant submits that the "rolling results window" of Figure 9 is a text-based interface, and simply displays results in a scrolling text display window, as is well known to those skilled in the art of programming.

Applicant further submits that the "rolling results window" of the results interface disclosed in Blowers is a feature of the runtime interface and is thus generic to any sequence/program created and run in this framework. In other words, the rolling results window is specifically not an automatically created graphical user interface that includes automatically created "graphical user interface elements associated with the one or more

parameters, wherein during execution of the program, the graphical user interface elements are displayed and are operable to receive user input and/or display output".

Said another way, as described in Blowers at col. 9, lines 16-25, the runtime screen shown in Figure 9, which is evidently displayed when a sequence is executed, is a standard predefined screen that is the same for every sequence that is executed. This runtime screen is not automatically (programmatically) generated during automatic generation of the program. Further, Blowers does not teach or suggest that automatic generation of the graphical user interface (which does not exist in Blowers) comprises automatically creating user interface elements associated with various parameters. Rather, Blowers evidently teaches a single generic runtime screen that is displayed with every sequence it executes. The runtime screen in Figure 9 of Blowers appears substantially like the sequence creation screens in Figures 7 and 8, except that a results interface 60 (not labeled in Figure 9) also evidently appears. According to Applicant's understanding of Blowers, this results interface window is generic to all sequences and is not programmatically generated based on parameters in the program being generated, and more specifically does not comprise automatically created graphical user interface elements that are associated with the parameters.

Applicant further notes the Blowers reference at column 9 lines 7-25 where the operation of saving and executing a desired sequence is discussed. As noted at column 9 line 13 "once the desired sequence has been created, it can be stored or saved in a condensed method within an inspection sequence file 52 which is usable by the engine 46." The paragraph in Blowers at column 9 lines 16-25 describes how the engine 46 executes the sequence file 52 utilizing a runtime screen as shown in Figure 9 of Blowers. Applicant respectfully submits that the runtime screen shown in Figure 9 does not satisfy the limitation of "wherein said programmatically generating the program comprises programmatically generating a graphical user interface for the program." Nor does Blowers teach the limitation "wherein, during execution of the program, the graphical user interface elements are operable to receive user input and/or display output".

In contrast, in Applicant's system, "the user may select from the displayed list the output parameters which are desired to be interactively viewable, and the prototyping

environment is operable to include a user interface control (in this case an indicator or display element) for viewing each selected output parameter value on the graphical user interface. The user interface control for a given input parameter is operable to receive user input, such as during program execution, and provide this input to the associated operation in the program. Similarly, the user interface control for a given output parameter is operable to receive output data from the program, such as during program execution, and display this output to the user" (page 8, line 25 through page 9, line 2). In other words, in Applicant's system, in automatically generating the graphical user interface for the program, specific GUI elements are automatically created, such as graphical indicators and controls, which are *associated* with the various parameters, and which facilitate user interaction with the selected parameters during program execution.

Thus, an important feature of the method of the present application is that a graphical user interface is automatically generated during automatic generation of a program. This automatic generation of the graphical user interface comprises automatically creating *graphical user interface elements* associated with parameters of the functional operations contained in the prototype created by the user, *wherein, during execution of the program, the graphical user interface elements are operable to receive user input and/or display output*. Applicant submits that this feature is simply not taught or suggested in Blowers.

The Office Action asserts that Blowers teaches automatically generating a graphical user interface for the program, citing col. 3, lines 15-16, and Figure 4. Applicant respectfully submits that the Examiner has incorrectly equated Blowers' interface for creating and specifying the program as disclosed in the cited passage and figure with Applicant's graphical user interface for interacting with the program during program execution. Applicant notes that Blowers' interface does not receive user input during program execution. In other words, Blowers' interface is specifically *not* for user interaction with the program during program execution.

Thus, Applicant respectfully submits that claim 1 and claims dependent thereon are patentably distinguishable over Blowers, and are thus allowable for at least the

reasons provided above. Independent claims 14, 27, and 40 include similar limitations as claim 1, and so the above arguments apply with equal force to these claims. Thus, for at least the reasons provided above, Applicant submits that claims 14, 27, and 40, and those claims respectively dependent thereon, are allowable.

Additionally, regarding dependent claims 8, 21, 34, and 44, which include the limitation "wherein the generated program is a graphical program, comprising a plurality of interconnected nodes that visually indicate functionality of the program", the Office Action asserts that the generated program of Blowers is a graphical program. Applicant respectfully disagrees.

Applicant respectfully submits that Blowers teaches away from this feature. Blowers teaches that the user is required to manually select graphical representations from the tool boxes of Figure 5 and place them into the tree structure of Figure 6 at a desired location. Thus, the user manually creates this tree structure. Moreover, Blowers repeatedly refers to "*interactively* developing a graphical control-flow structure and associated application software" (Title, independent claims, and specification), and so specifically limits the claimed invention to this interactive approach. For example, as mentioned above, Blowers discloses (in col. 8, lines 64-67) "Graphical representations or icons are selected from the tool boxes of FIG. 5 which correspond to desired functional tasks and are linked into the tree structure of FIG. 6 by a task sequencer interface 50 in the desired locations."

Thus, Blowers teaches away from the present claims in that Blowers teaches that a user is required to manually or interactively develop the "graphical control flow structure" by selecting these icons and placing them at desired locations in the tree structure of Figure 6. In other words, Blowers specifically does not teach automatic generation of a graphical program. Applicant thus submits that for at least these reasons, claims 8, 21, 34, and 44 are patentable distinct from Blowers, and are allowable.

Thus, Applicant respectfully submits that the present claims are allowable over the Blowers reference, and the removal of the 102 rejection of claims 1, 3-5, 7-14, 16-18, 20-27, 29-31, 33-40, and 42-45 is respectfully requested.

Applicant further submits that new claims 46 and 47 include further limitations not taught or suggested by Blowers, specifically, "automatically generating a block diagram, wherein the block diagram comprises a plurality of interconnected nodes that visually indicate the functionality of the program", and "automatically generating a user interface panel, wherein the user interface panel comprises the graphical user interface elements", respectively, and so are also allowable.

Section 103 Rejections

Claims 6, 19, and 32 were rejected under 35 U.S.C. 103(a) as being unpatentable over Blowers. Applicant respectfully traverses the rejection.

Amended claim 6 recites:

6. The method of claim 1, wherein a plurality of parameters are associated with the functional operations, the method further comprising:

receiving user input specifying which of the plurality of parameters are desired to have associated graphical user interface elements;

wherein said generating the graphical user interface comprises creating a graphical user interface element associated with each specified parameter, but not creating graphical user interface elements associated with unspecified parameters.

The Office Action asserts that Blowers discloses receiving user input specifying which of the plurality of parameters are desired to have associated user interface controls, citing col. 3, lines 28-31, and the Blowers further discloses that the generated graphical user interface comprises creating user interface controls associated with each specified parameters, citing col. 3, lines 31-35. Applicant respectfully disagrees.

The cited passages describe user specification of machine vision tasks and hardware which are graphically represented by nodes in a tree structure, and user configuration of parameters associated with them. Nowhere in the cited passages (or elsewhere in Blowers) are graphical user interface elements mentioned or even hinted at,

and more specifically, where the graphical user interface elements are associated with each specified parameter, and operable to receive user input and/or display output during execution of the program.

Moreover, as argued above, independent claims 1, 14, 27, and 40, have been shown to be allowable, thus claims 6, 19, and 32, which are respectively dependent from claims 1, 14, and 27, are patentably distinct and non-obvious over Blowers, and are thus also allowable, for at least the reasons provided above.

Removal of the 103 rejection of claims 6, 19, and 32 is respectfully requested.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-43700/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☒ Notice of Change of Address

☒ Request for Continued Examination

Respectfully submitted,

Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 4/21/2004